FSTaxis Algorithm: Bio-Inspired Emergent Gradient Taxis

Joshua Cherian Varughese[1,2], Ronald Thenius[1], Franz Wotawa [2] and Thomas Schmickl[1]

[1]Artificial Life Lab, Department of Zoology, Karl-Franzens-Universität Graz
[2]Institute for Software Technology, Techische Universität Graz
joshua.varughese@uni-graz.at

Abstract

This article presents a novel bio-inspired emergent gradient taxis principle for robot swarms. The underlying communication method was inspired by slime mold and fireflies. Nature showcases a number of simple organisms which can display complex behavior in various aspects of their lives such as signaling, foraging, mating etc. Such decentralized behaviors at the organism level gives rise to an emergent intelligence such as in bees, slime mold, fireflies etc. Chemo taxis and photo taxis are known to be abilities exhibited by simple organisms without elaborate sensory and actuation capabilities. Our novel algorithm combines the underlying principles of slime mold and fireflies to achieve gradient taxis purely based on neighbor-to-neighbor communication. In this article, we present a model of the algorithm and test the algorithm in a multiagent simulation environment.

Introduction

Swarm robotics research has shown that complex problems can be solved in unconventional ways (Schmickl and Hamann, 2011)(Schmickl et al., 2008)(Bjerknes et al., 2007). Swarm intelligence uses simple agents following simple rules to solve complex problems. Without the advantages of swarm intelligence, such complex problems could only be solved at a relatively high amount of computational and economic resources. In project subCULTron (subCULTron, 2015), we aim to develop an underwater society of learning, adapting, self-sustaining robots which can be used for various applications. Given the challenges of robotic systems underwater such as limited availability of classical communication systems, limited mobility etc., there need to be stable but simple solutions. In nature, there are many such behaviors such as chemical gradient taxis (chemo taxis) in slime mold (Webb, 1998) or thermal taxis in bees (Grodzicki and Caputa, 2005). It is challenging that nature solves these problems with the minimum resources. Gradient taxis is an example of an algorithm that will be used in subCULTron for gradient ascent or descent. Like many swarm researchers in the past (Schmickl and Crailsheim, 2007) (Nakagaki, 2001), we draw inspiration from nature to solve the gradient ascent for robots in subCULTron.

Many studies in the past have been done on application of swarm based algorithms in robotics. Swarm behavior is based on decentralized underlying rules at the organism level giving rise to an emergent intelligence such as in bees (Bodi et al., 2015) (Kernbach et al., 2009), slime mold (Nakagaki et al., 2004), fireflies (Buck and Buck, 1966) etc. The aggregation and maze solving capabilities of slime mold have been extensively researched (Nakagaki, 2001). Slime mold swarms have been used to solve mazes (Nakagaki et al., 2000) and this capability has been tested in real world scenarios such as the Tokyo rail transport system (Nakagaki et al., 2004). Similarly, fireflies and their ability of phase synchronized pinging has been of interest to the research world for a long period of time (Buck and Buck, 1966). Many of such behaviors has found applications in engineering and computer science. For example, Yang (2009) has taken inspiration from fireflies to solve multi-modal optimization problems and such efforts have shown promising results.

In this paper, we present a novel method which combines communication behavior from slime mold as well as fireflies for gradient ascent. Various kinds of gradient functions to a swarm of agents for testing the algorithm. The following sections will first formulate the algorithm, describe the testing scenarios, methods and discuss the results in that order. Our algorithm is a fine example of how emergent solutions can be used for tasks without using complex computation, large amount of memory and with minimum power consumption. There exist classical approaches for gradient ascent and multi-modal optimization such as the steepest gradient descent (Arfken, 1985), Particle Swarm Optimization (Kennedy and Eberhart, 1995) etc. Another possible solution for gradient related problems is Simultaneous Localization And Mapping (SLAM)(Bazeille and Filliat, 2011) where an agent constructs a map of an unknown environment while simultaneously keeping track of its own location and the gradient value. Then a global observer is able to guide the agent finally to the maximum or minimum gradient value. However, these solutions require high amount of computation power which is economically and computation-

wise unfeasible for a swarm of underwater robots such as that in subCULTron.
The objectives of this paper are as follows:

1. Formulate a novel emergent gradient taxis algorithm inspired by slime mold and fireflies.
2. Test the algorithm in various types of gradients.
3. Validate the algorithm by investigating boundary conditions.
4. Discuss strengths and weaknesses of the algorithm and compare it with an existing emergent gradient taxis algorithm (SwarmTaxis Bjerknes et al. (2007)).

Biological Inspiration

As previously mentioned, the presented algorithm takes inspiration from slime mold and fireflies. Therefore, it is of merit to look into the aspects of their biological behavior that we draw inspiration from. This section briefly discusses communication strategies used by slime mold and fireflies which will support the formulation of the algorithm.

Slime Mold

Slime mold (*Dictyostelium Discoideum*), is a free living diploid life form. It has been subject of much study in the past due to its ability to survive harsh environments by taking advantage of group behavior. Slime mold, during its life cycle, aggregates with other cells to from a multicellular organism. Each organism starts its life as a unicellular amoeba, but during starvation they aggregate to form a multi cellular fruiting body. Chisholm and Firtel (2004) divide its life cycle as follows: Aggregation, Streaming, Slug, Culmination, Fruiting body. The algorithm presented in this paper will deal mainly with the aggregation phase and hence will look it in detail.

When there are ample food sources, cells grow and divide in a matter of three to four hours (Siegert and Weijer, 1992). On the other hand, if there is a scarcity of food, significant cooperation between the cells begin, thereby kicking off the aggregation phase. During this time, some cells (centers) release Cyclic Adenosine Monophosphate (cAMP) into the environment to induce a chemical concentration spike around them (Siegert and Weijer, 1992). cAMP concentration diffuses very quickly into the environment and therefore the chemical spike is short-lived. This chemical spike enables these centers to recruit other cells present around them. When surrounding cells perceive this chemical signal, they move towards areas of high cAMP concentration and release cAMP themselves, thereby relaying the signal. This in turn, attracts other cells towards the centers. One cell is able to release cAMP at an interval of 12-15 seconds (Alcantara and Monk, 1974); during this interval, individual cells are insensitive to cAMP pulses. This interval can be understood as the refractory phase of the amoeba. The signal relaying mechanism described above forms the basis for spatiotemporal patterns known as scroll waves (Siegert and Weijer, 1992). The refractory phase is responsible for these scroll waves as it prevents the signaling organism from perceiving its own signal that was relayed earlier. The emergence of scroll waves enable the amoeba to move towards the recruiting centers for successful aggregation.

Fireflies

Fireflies are a family of insects that are capable of producing bio-luminescence to attract a mate or a prey (Buck and Buck, 1966). The brightness of the bio-luminescent light depends on the amount of luciferin, a light emitting compound, available with the firefly (de Oliveira et al., 2011). Bio-luminescence of various families of fireflies has been a subject of elaborate study in the past (Buck and Buck, 1966). Apart from being able to blink, fireflies are known to behave in cooperation with other fireflies. It is a spectacular sight to see thousands of fireflies light up in unison on a tree lighting it up entirely. This uniform blinking is in order for the swarm to have higher chance of attracting mates or prey (Buck and Buck, 1966). The luminescence of the blinking swarm is much more than that of an individual firefly.

Such synchronicity is a result of a simple mechanism by which initially the individual fireflies blink randomly and when it perceives a blink in its surrounding, it blinks again and then resets its own frequency to match the other (Camazine et al., 2001). It takes time for the fireflies to achieve complete synchronization. This is analogous to a phase couple oscillator which adjusts its phase to match it to that of the faster one in the vicinity. This trait emerges into a pseudo synchronized blinking pattern while the frequency of blinking will be influenced by the fastest blinking insect.

FSTaxis algorithm

As per the objectives listed in the introduction, a novel gradient taxis algorithm is hereby presented, namely, the Firefly Slime mold Taxis(FSTaxis) algorithm. As its name suggests, this algorithm draws inspiration from biological systems introduced in the Section "Bio Inspiration". The FSTaxis algorithm makes use of the communication strategy of slime mold and the phase coupled oscillation aspect in fireflies. The behavior of agents in the FSTaxis algorithm can be broadly classified into *Ping behavior* and *Motion behavior*. The following sections will explain the working of these behavior modes. The sequential flow of instructions of the FSTaxis algorithm can be found on the following page. Hereafter, a "ping" is referred to the single bit communication which each agent broadcasts. The agents are equipped with sensors to determine the direction of incoming pings and the environmental factor of interest value at its own location.

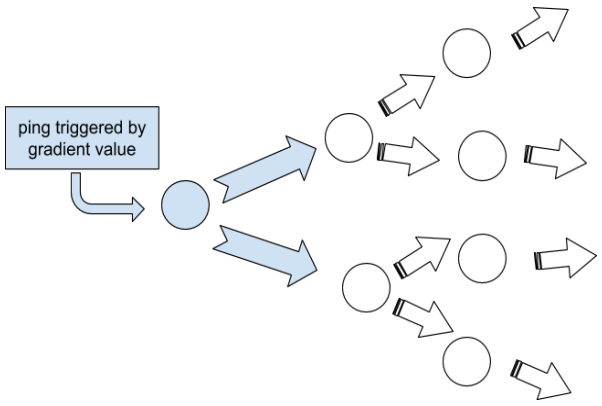

Figure 1: When the intrinsic cycle length of each agent counts out, a ping is broadcasted, surrounding agents captures the ping and relays it on. The blue circles show original agent whose internal clock triggered. The white agents in the figure relay the pings

.

The pings can be perceived by other robots within a very limited sensor radius, s_r. Also, the agents are able to move around in the environment with limited speed v_a.

Ping behavior

Each agent has three communication states: "pinging", "refractory" and "inactive" as shown in the state transition diagram Figure 2. By default all agents are set to inactive mode and each of them have an internal countdown timer whose initial value is associated with its position in the environmental gradient. In the inactive mode, the agent only checks for incoming pings. When an agent receives a ping, it broadcasts a ping for a period of time, say t_p. During t_p, the agent is said to be pinging and after t_p, the agent enters the refractory mode. During refractory time, the agent ignores all incoming pings. After the refractory time t_r, the agent sets itself back to inactive mode. The cycle continues if the agent receives another ping.

Each agent has an inherent cycle time determined by the environmental gradient at its position. As shown in Figure 1, if the internal timer of any agent counts to zero before a ping is received, the agents broadcast a ping and sets its own ping frequency, f_p , by associating it with the gradient value at its position, g_p. That "original" ping is further relayed by the neighboring agents as per the ping behavior explained above. The agent that triggers the original ping (the agent whose f_p counted to zero) is referred to as the "leader" in the upcoming sections of this paper.

In order to provide scaling of pinging frequencies to meaningful values, two preset maximum and minimum are selected for the gradient under consideration. Let these values be g_{max} and g_{min}. Equation 1 shows the relation between ping frequency of agents and inherent cycle time of agents. The selection of constants, α and ω, are dependent upon t_p, t_r and the boundary values of the gradient under consideration. Here, α and ω should be selected so that the agents continue pinging in the entire range of gradient values possible. For example, if t_p is equal to f_p for any agent, it will continuously ping without ever entering refractory phase. Therefore, it is necessary that α and ω are adjusted to scale ping frequencies to meaningful values. In this paper, α and ω are selected only to demonstrate the gradient ascent ability of the FSTaxis algorithm. Since it does not depend upon the type of gradient the values of the constants will be the same throughout this paper as shown in Table 1.

$$f_p = \alpha + \frac{((g_p - g_{min})}{(g_{max} - g_{min})} * \omega \tag{1}$$

Motion behavior

Motion behavior in the agents is overseen by ping behavior. An agent in inactive mode does not move. As shown in Figure 2, motion is initiated in the active mode. When any agent receives a ping it sets itself to active mode, sets its own heading towards the received ping and starts moving to cover a fixed distance, β at velocity v_a. A ping can only be perceived within the limited sensor range, s_r, of the robot, therefore it limits the number of agents that are able to affect any particular agent. In the scenario described above, it is possible that each agent receives multiple pings from different directions, h_n, where n is the number of agents pinging. In such as case, the agent will calculate the mean heading, h_{mean}, and set its heading towards this mean.

If an agent's internal clock triggers, an "original" ping based on the environmental value is broadcast; then, this agent labels itself the leader and does not move in that particular cycle.

When a swarm of agents execute the FSTaxis algorithm as per description above, scroll waves of pings similar to that in slime mold (as mentioned in section "Slime mold") propagates through the swarm. Since the internal timer of the robot at highest gradient value will count to zero first, the direction of the wave will be from the higher to lower gradients. During their inactive cycles, the agents will move towards the mean direction of incoming pings. Since the "leader" broadcasts the original ping and does not move, the agents will gather around the leader. When the agents are in their new position, their internal clock takes the values of the environmental value (gradient value). Whichever agent's internal clock triggers first becomes the leader and the swarm then gathers around this agent. This repeated choosing of leaders and gathering around the leader will draw the swarm towards areas with higher gradient value and in essence emerges into a gradient ascent.

Method

To demonstrate the gradient ascent capability, a linear gradient, a hyper ellipsoid gradient and noisy variants of these

Algorithm 1 The FSTaxis algorithm

repeat
  procedure PING BEHAVIOR(t_p, t_r,v_a,t_f)
    for all agents **do**
      if pingmode = refractory mode **then**
        count down t_r
        if $t_r = 0$ **then**
          set state ← inactive mode
          set $t_f \leftarrow 1/f_p$ - $equation1$
          set leader status ← "OFF"
        end if
      end if
      if pingmode = active state **then**
        count down t_p
        if $t_p = 0$ **then**
          set state ← refractory mode
        end if
      end if
      if pingmode = inactive mode **then**
        if any ping received? **then**
          set state ← active mode
          set move agent ← "ON"
        end if
      end if
      count down t_f
      if $t_f = 0$ **then**
        set state ← active mode
        set leader status ← "ON"
      end if
    end for
  end procedure
until $forever$

repeat
  procedure MOVEMENT(move agent, leader status)
    for all agents with movement = "ON" and leader status ! = "ON" **do**
      while $distancecovered < \beta$ **do**
        Create empty list, l
        for $i \leftarrow 1, no : ofpingsreceived$ **do**
          append list $l \leftarrow h_i$
        end for
        calculate h_{mean} of list, l
        set agent heading $h_a \leftarrow h_{mean}$
        move agent with fixed velocity v_a
      end while
      set move agent ← "OFF"
    end for
  end procedure
until $forever$

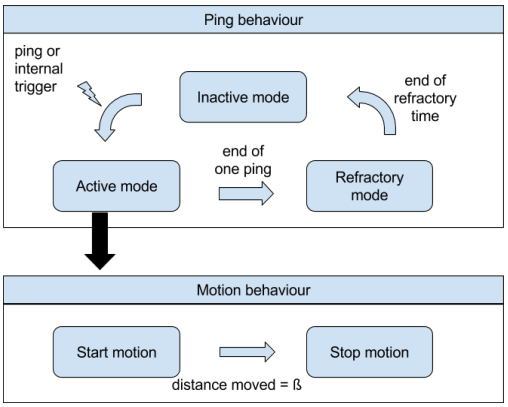

Figure 2: Figure shows the state transition diagram of the FSTaxis algorithm. The algorithm has two behaviors - ping behavior and motion behavior. In ping behavior, there are three states: active, refractory and inactive. An agent is in active state when it receives a ping from a surrounding agent or when its own internal clock triggers. After the ping duration, the agent transitions into a refractory mode. After the refractory time, the agent transitions into the inactive mode. The active mode triggers motion behavior and the agent takes a preset distance towards the ping it received.

gradients are used as test functions. Since depth is of relevance in project subCULTron, it is considered to be the physical quantity of interest in the paper. As mentioned before, the frequencies are scaled according to the equation 1 and Table 1 shows all the constants used in this experiment. The simulation environment used is Netlogo 4.3.1 (Wilensky, 1999). In Netlogo, the test area is divided into "patches" (spatial units) and the agents are called "turtles". For the purpose of this experiment, depth is the physical quantity associated with each patch. The sensor radius of each of the agents are measured in patches and in this experiment it is taken to be 3 patches since it is a reasonable range for underwater communication.

Constants								
	ω	t_p	t_r	s_r	β	α	g_{max}	g_{min}
Value	0.1	5	5	3	1.5	0.008	50	5
Units	-	s	s	p[1]	p[1]	-	m	m

Table 1: Table showing all constants used in the FSTaxis algorithm.

Linear gradient

This section aims to demonstrate the basic implicit capability of the FSTaxis algorithm to traverse gradient. The equation for gradient is that of a line, as shown in equation 2

$$f(x) = x \tag{2}$$

[1]unit p in Table 1 represents number of patches in netlogo

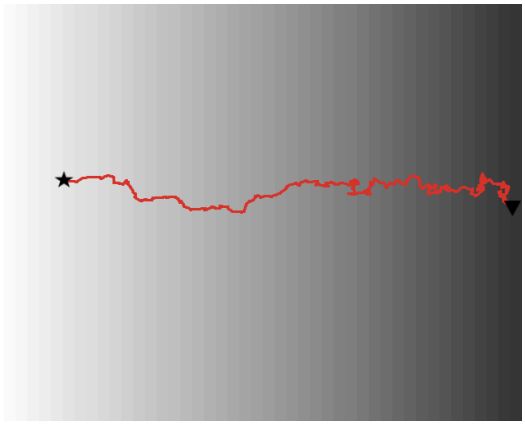

Figure 3: A linear gradient was presented to the FSTaxis algorithm. The grayscale shows the gradient value and the goal is the darkest area; the star symbol represents the starting point and the inverted triangle marks where the swarm converged.

Figure 3 shows the simulation environment in Netlogo setup with a linear gradient. The color scaling represents the gradient value of the environment and therefore the goal of the gradient ascent algorithm will be to go towards the dark colored areas. The red line represents the trajectory of the mean position of the swarm from starting point, represented by the star, to convergence, represented by the inverted triangle. The swarm is said to converge when its mean position oscillates around the area with the maximum gradient value. The trajectory shown is the result of one of the exemplary run from the 100 runs conducted with this test gradient. 100% of the runs resulted in convergence to maximum gradient value.

Hyper ellipsoid gradient

This section presents an environment for testing the FSTaxis algorithm in a relatively more complex gradient, a three dimensional axis parallel hyper ellipsoid. The axis parallel hyper ellipsoid, represented by its standard equation 3, is a convex, continuous function and multiple modal function.

$$f(x) = \sum_{i=1}^{2} x_i^2, where - 5.12 < x_i < 5.12 \qquad (3)$$

For the hyper ellipsoid gradient, there are four goals at the corners of the arena with the highest gradient value. The area of the goal (corners of the ellipsoid) is merely 0.23% of the total area of the arena. Therefore, random chances of the swarm converging to the goal is minimal. Figure 4 shows FSTaxis algorithm tested with a hyper ellipsoid gradient, the thick red line shows the trajectory of the swarm. The black star marks the starting point and the inverted triangle shows the area of convergence. The trajectory shown is the result of one of the exemplary run from the 100 runs conducted with this test gradient. 100% of the runs resulted in convergence to maximum gradient value.

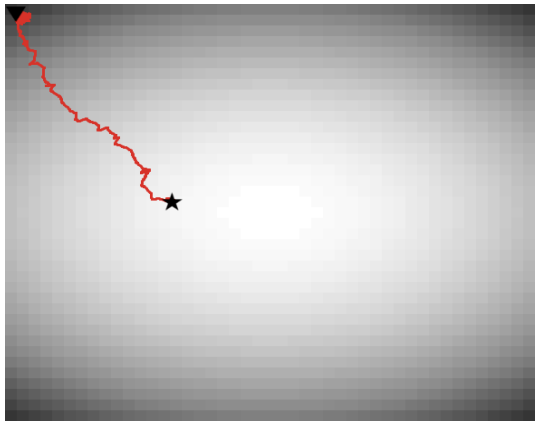

Figure 4: An axis parallel hyper ellipsoid gradient was presented to a swarm executing FSTaxis algorithm. The grayscale shows the gradient value and the goal is the darkest area; the star symbol represents the starting point and the inverted triangle marks where the swarm converged.

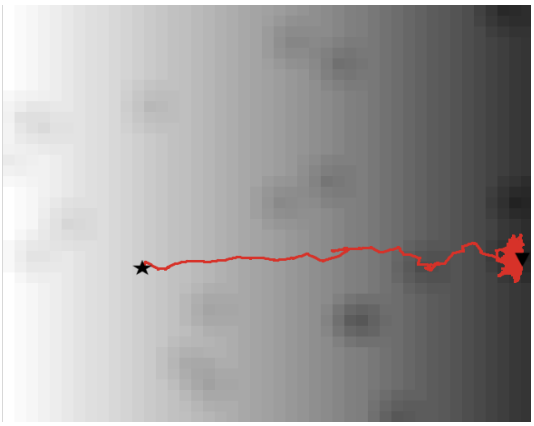

Figure 5: A linear gradient with numerous local maxima was presented to the FSTaxis algorithm. The grayscale shows the gradient value and the goal is the darkest area; the star symbol represents the starting point and the inverted triangle marks where the swarm converged.

Noisy Gradients

In order to test the ability of the FSTaxis algorithm to overcome small local optima, 20 randomly generated obstructions or "hills" have been introduced to the smooth gradient. These obstacles attracts them to stay at these local optima if sufficient exploration is not introduced. Figure 6 and 5 shows the result of a random successful attempt out of the 10,000 iterations of FSTaxis algorithm run with noisy hyper ellipsoid gradient and linear gradient respectively. The experiments with noisy gradients were tested with varying steepness of local optima and spread of each optima. The number of obstructions were kept constant at 20. For each obstruction spread ranging from 1 to 10 and steepness ranging from 0.25 to 2.5 times the normal gradient, 100 iterations were run to observe the convergence to the goal.

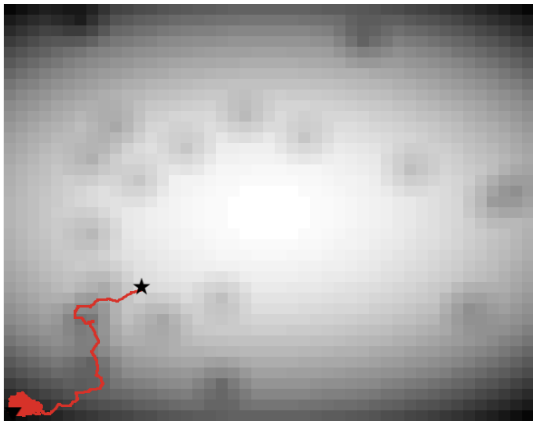

Figure 6: A axis parallel hyper ellipsoid gradient with numerous local maxima was presented to the FSTaxis algorithm. The grayscale shows the gradient value and the goal is the darkest area; the star symbol represents the starting point and the inverted triangle marks where the swarm converged.

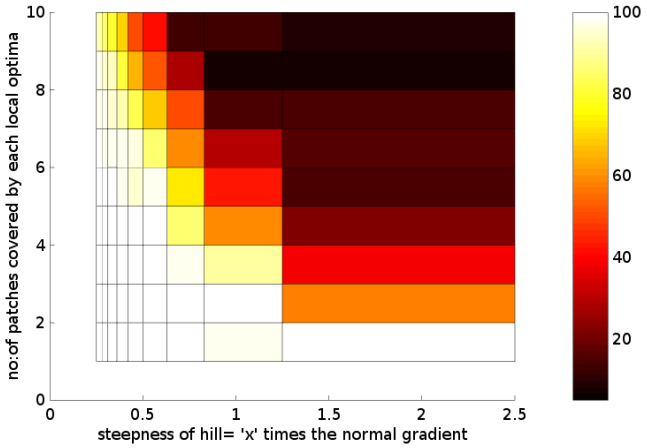

Figure 7: The color map shows the percentage convergence in presence of noise. Each gradient was introducted with 20 obstructions or local optima. The X-axis represents the steepness of each of these obstructions with respect to the normal gradient. The Y–axis represents the area covered by each of these 20 obstructions(measured in patches). Each patch is 0.05% of the entire arena. For each set of hillsize and steepness, 100 iterations were run. White coloured areas show 100% convergence and shades towards red and black coloured areas shows lower convergence as per color map provided.

Results

The FSTaxis algorithm, as described in the section "Methods", successfully traverses the linear gradient as well as the hyper ellipsoid gradient in all 100 iterations conducted. Figures 3 and 4 show clearly the ability of the algorithm for gradient ascent.

Performance in noisy gradients

When the agents executing FSTaxis algorithm (see subsection "Noisy Gradients") are presented with obstructions in the gradient, they are able to overcome local optima introduced. It can be seen that the agents are eager to climb gradients as seen in 5 and 6. As individual agents move towards the leader, they overshoot the leader (agent whose internal clock triggered a ping) and cross out of the hill to escape the local optima. Figure 7 is a graph relating between the size and steepness of local optima (number of patches) to the percentage runs that converged to the goal. It is seen that when size of local optima is under three patches (for 20 obstacles in the arena) 100% of runs converge to the goal. As size and spread of local optima rises, the rate of convergence decreases.

Performance with multiple gradients

Figure 8 shows the region of convergence relative to the starting point of the swarm for 100 iterations. The X-axis shows the quadrant in the arena where the swarm started and the Y-axis shows the region of convergence. Numbers 1, 2, 3, 4 refer to the quadrants as referred to in the cartesian coordinate system. It is seen that in 95% of the runs, the swarm converges to the goal nearest to it. The 5 % error is attributed to the fact that, when the swarm starts at the midpoint between two gradients, it has to choose which gradient to ascent. This decision is taken randomly since it depends on which agent's internal clock triggers first.

Discussion

In contrast to many classical solutions of gradient ascent, the FSTaxis algorithm is efficient, simple and requires only simple hardware. Since agents do not actively compare its current gradient value and the previous gradient value, gradient ascent is purely an emergent trait. The FSTaxis algorithm uses no evaluation function and acts purely based on local knowledge. Any agent has to be merely informed about the presence of other agents in its sensor radius. Therefore, the impositions on the agent is to sense the gradient value at current position, adjusting the agent's own ping behavior accordingly and broadcast a 1-bit communication to make its presence known. These qualities make FSTaxis algorithm a choice solution for real world gradient taxis when resources are sparse. Many gradient taxis solutions have been proposed in the past for single agents. Some examples include the standard hill climber (Davis, 1991), helical klinotaxis (Long et al., 2004) etc. While these solutions work well for individual robots, they are not designed for a group of robots. There are a few swarm algorithms for emergent gradient taxis that has been proposed in the past like swarmtaxis(Bjerknes et al., 2007), Artificial Homeostatic Hormone System (Schmickl et al., 2010) etc. The swarmtaxis algorithm is an emergent gradient taxis solution like the FSTaxis algorithm and is based on a single ping broadcast communication. Therefore, it is of merit to compare the FSTaxis

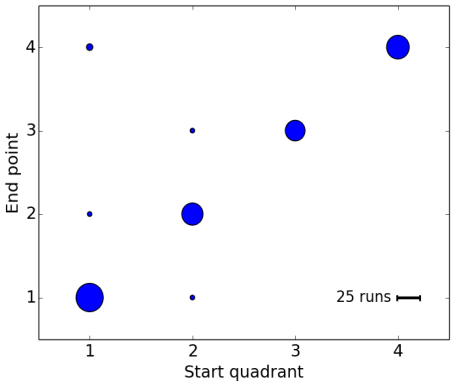

Figure 8: Bubble plot showing the region of convergence relative to the starting point of the swarm for 100 iterations. The reference line at the right lower corner shows the diameter of a 25 run bubble. The X-axis shows the quadrant in the arena where the swarm started and the Y-axis shows the region of convergence. Numbers 1, 2, 3, 4 refer to the quadrants as referred to in the cartesian coordinate system. It is seen that in 95% of the runs, the swarm converges to the goal nearest to it.

algorithm with the swarmtaxis algorithm.

Bjerknes et al. (2007) presented the "swarmtaxis" algorithm which is also an emergent solution for gradient taxis. It is worthwhile to mention swarmtaxis as the FSTaxis algorithm drew inspiration from this work. The swarmtaxis algorithm works by creating a frontier of agents which are facing the source (light) terming them "illuminated". The illuminated agents cast a shadow on agents behind them terming them as "shadowed" agents. The swarmtaxis algorithm works based on the illuminated robots having a higher sensing distance than the shadowed robots and hence, they move away from the shadowed robots. In essence, they move towards the light source.

The swarmtaxis algorithm guarantees that the swarm will converge to the source. It is a stable way to ascent the light gradient however it is not suitable for use in swarm robotics as it imposes various limitations on the kind of gradient it can ascent. For example, the swarmtaxis algorithm assumes that each robot has the ability to physically occlude another from the source. This assumption holds well when light source is at the same level as the agents and not otherwise. FSTaxis algorithm overcomes this limitation by being dependent upon the local gradient value. Therefore, a need for occlusion never arises.

If there are two light sources, the swarmtaxis algorithm will create two frontiers and will not consider the brighter of the sources to move to. Although the swarmtaxis algorithm makes it redundant to have a local gradient value sensor, it is at the cost of not being able to handle multiple sources. The FSTaxis algorithm, on the other hand, is able to handle multiple sources or maxima at the cost of using a local gradient value sensor.

As per the objectives of the paper, boundary conditions for FSTaxis algorithm were investigated. It has been shown that the FSTaxis algorithm is able to work with multiple local optima. As seen in Figure 7, in presence of local optima that are steep enough, the FSTaxis algorithm is likely to get stuck in the local maxima. For problems such as multimodal optimization, for example solving a rotated hyper ellipse, the FSTaxis algorithm does not guarantee convergence to the global maxima.

Conclusion

From this paper, it is demonstrated that FSTaxis algorithm is a feasible solution for gradient ascent in swarm robotics. It is especially attractive because it requires only a single bit communication between the agents.

As discussed previously, the FSTaxis algorithm does not guarantee a solution for multi-modal gradients. In the future, extensions of this algorithm can be formulated for use in multi-modal optimization.

The FSTaxis algorithm works based on sharing of spatial gradient related information via frequency of pings. This paper is a successful demonstration of information exchange without explicitly sending data. In the future, more resources can be dedicated towards how more information regarding gradients can be shared and how the agents can use this to change their behavior.

For relating the current position with its own ping frequency frequency, agents executing the FSTaxis algorithm scales the gradient value as per equation 1. This provides a way to tweak the equation depending on the gradient that is of interest. Moving forward, it would be useful to have a single equation that can be used globally without a need to manually scale the parameters.

Acknowledgments

This work was supported by EU-H2020 Project no. 640967, subCULTron, funded by the European Unions Horizon 2020 research and innovation programmer under grant agreement No 640967.

References

Alcantara, F. and Monk, M. (1974). Signal propagation during aggregation in the slime mould dictyostelium discoideum. *Microbiology*, 85(2):321–334.

Arfken, G. (1985). The method of steepest descents. *Mathematical methods for physicists*, 3:428–436.

Bazeille, S. and Filliat, D. (2011). Incremental topo-metric slam using vision and robot odometry. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4067–4073. IEEE.

Bjerknes, J. D., Winfield, A., and Melhuish, C. (2007). An analysis of emergent taxis in a wireless connected

swarm of mobile robots. In *IEEE Swarm Intelligence Symposium*, pages 45–52, Los Alamitos, CA. IEEE Press.

Bodi, M., Möslinger, C., Thenius, R., and Schmickl, T. (2015). {BEECLUST} used for exploration tasks in autonomous underwater vehicles. *IFAC-PapersOnLine*, 48(1):819 – 824. 8th Vienna International Conferenceon Mathematical ModellingMATHMOD 2015.

Buck, J. and Buck, E. (1966). Biology of synchronous flashing of fireflies. *Nature*, 211:562–564.

Camazine, S., Denenbourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2001). Synchronized flashing among fireflies. pages 143–166. Princton University Press, Princton.

Chisholm, R. L. and Firtel, R. A. (2004). Insights into morphogenesis from a simple developmental system. *Nature reviews Molecular cell biology*, 5(7):531–541.

Davis, L. (1991). Bit-climbing, representational bias, and test suite design. In *ICGA*, pages 18–23.

de Oliveira, D. R., Lopes, H. S., and Parpinelli, R. S. (2011). *Bioluminescent swarm optimization algorithm*. INTECH Open Access Publisher.

Grodzicki, P. and Caputa, M. (2005). Social versus individual behaviour: a comparative approach to thermal behaviour of the honeybee (*Apis mellifera* l.) and the american cockroach (*Periplaneta americana* l.). *Journal of Insect Physiology*, 51(3):315 – 322.

Kennedy, J. and Eberhart, R. C. (1995). Particle swarm optimization. In *IEEE International Conference on Neural Networks*, Los Alamitos, CA. IEEE Press.

Kernbach, S., Thenius, R., Kornienko, O., and Schmickl, T. (2009). Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic swarm. *Adaptive Behavior*, 17:237–259.

Long, J. H., Lammert, A. C., Pell, C. A., Kemp, M., Strother, J. A., Crenshaw, H. C., and McHenry, M. J. (2004). A navigational primitive: biorobotic implementation of cycloptic helical klinotaxis in planar motion. *IEEE Journal of Oceanic Engineering*, 29(3):795–806.

Nakagaki, T. (2001). Smart behavior of true slime mold in a labyrinth. *Research in Microbiology*, 152:767–770.

Nakagaki, T., Yamada, H., and Hara, M. (2004). Smart network solutions in an amoeboid organism. *Biophysical Chemistry*, 1:1–5.

Nakagaki, T., Yamada, H., and Toth, A. (2000). Intelligence: Maze-solving by an amoeboid organism. *Nature*, 407:470.

Schmickl, T. and Crailsheim, K. (2007). A navigation algorithm for swarm robotics inspired by slime mold aggregation. In Şahin, E., Spears, W. M., and Winfield, A. F. T., editors, *Swarm Robotics - Second SAB 2006 International Workshop*, Lecture Notes of Computer Science, pages 1–13. Springer-Verlag, Berlin, Heidelberg, New York.

Schmickl, T. and Hamann, H. (2011). BEECLUST: A swarm algorithm derived from honeybees. In Xiao, Y., editor, *Bio-inspired Computing and Communication Networks*. CRC Press.

Schmickl, T., Hamann, H., Stradner, J., Mayet, R., and Crailsheim, K. (2010). Complex taxis-behaviour in a novel bio-inspired robot controller. In *Proc. of the ALife XII Conference*, pages 648–655. MIT Press.

Schmickl, T., Thenius, R., Möslinger, C., Radspieler, G., Kernbach, S., and Crailsheim, K. (2008). Get in touch: Cooperative decision making based on robot-to-robot collisions. *Autonomous Agents and Multi-Agent Systems*, 18(1):133–155.

Siegert, F. and Weijer, C. J. (1992). Three-dimensional scroll waves organize dictyostelium slugs. *PNAS*, 89(14):6433–6437.

subCULTron (2015). Submarine cultures perform long-term robotic exploration of unconventional environmental niches. http://www.subcultron.eu/.

Webb, B. (1998). Robots, crickets and ants: models of neural control of chemotaxis and phonotaxis. *Neural Networks*, 11:1479–1496.

Wilensky, U. (1999). Netlogo. *Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.*

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *Stochastic algorithms: foundations and applications*, pages 169–178. Springer.